Choix technique de développement

# Choix de la technologie pour le projet MADERA

## Type d’application

Contrainte principale :

L’application pour l’entreprise MADERA doit être utilisable sur tablette et ordinateur ainsi que disponible hors connexion.

Nos possibilités sont donc :

### Client léger :

**Avantages :**

* Mise à jour centralisé du site. Le déploiement de MaJ se fait facilement et rapidement.
* Accessibilité.
* Peu sensible aux pannes matérielles.

**Inconvénient :**

* Langages client/serveur différents
* Partie cliente nécessitant plusieurs langages
* Evolutions des navigateurs
* Les performances de l’application peuvent baisser s’il a un trop grand nombres utilisateurs simultanés.
* **Ne permet pas le hors connexion.**

### Client Lourd :

**Avantages :**

* Un seul langage pour les parties client et serveur
* Interaction avec le serveur simplifié.
* **Hors connexion possible.**

**Inconvénients :**

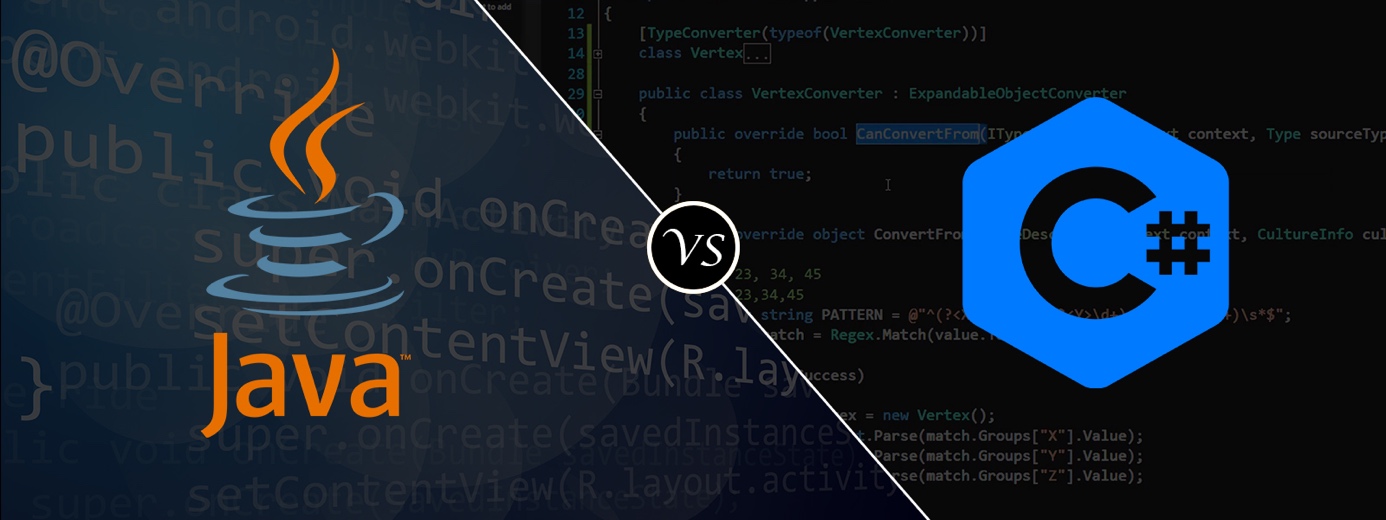
* Déploiement plus lourd
* Maintenance et Mises à jour plus lourdes
* Sécurité et sauvegardes dépendantes du poste client
* Performances en baisse si trop nombreux utilisateurs.
* Formation de l’équipe a la création de client lourd.

### Conclusion

L’obligation d’une utilisation hors connexion de l’application nous impose le choix d’un client lourd.

## Langage de programmation

Pour un client lourd tel que l’application que nous allons réaliser durant le projet MADERA, deux langages sortent du lot.



Le langage Java et le C# sont très clairement similaires et leurs performances globales pour la réalisation d’un projet tel que celui-ci se rejoignent.

Java et C# partagent un ensemble de fonctionnalités communes :

* Compilation dans un langage intermédiaire indépendant de la machine et exécution dans un environnement dédié (une machine virtuelle)
* Gestion automatique de la mémoire grâce à un ramasse-miettes
* Introspection pour manipuler dynamiquement les objets
* Toutes les classes héritent d'une même classe (Object) et sont allouées sur le tas
* Pas de support de l'héritage multiple mais utilisation d'interfaces
* Tout doit être encapsulé dans une classe : il n'existe pas de fonctions ou constantes globales
* Gestion des erreurs grâce aux exceptions

Notre principal facteur de décision pour nous est donc la connaissance du langage par les membres de l’équipe. Notre équipe ne possède pas de membre ayant réalisé de client lourd C# ou Java par le passé. Il faudra donc se former sur le langage pour réaliser ce projet.

|  |  |  |
| --- | --- | --- |
|  | Java | C# |
| Connaissance du langage | 1/10 | 4/10 |

**Nous avons donc fait le choix du C# pour la réalisation de l’application.**

Pour l’interface graphique de notre application, nous utiliserons XAML pour sa compatibilité avec le langage C#. Nous permettant de séparer les développements de l’équipe en deux parties.

L’une sur les traitements avec le serveur et la seconde sur la conception graphique de l’application.

# Logiciel de développement

Le logiciel de développement que nous avons choisis d’utiliser est Visual studio. Nous l’utiliserons car tous les projets de l’entreprises pour ce genre d’applicatif lourd est réalisé dessus. Nous avons donc les licences à disposition.

De plus il permet de générer des applications C#/XAML sans soucis.



# Système de gestion de base de données relationnelle

Dans le cadre du projet, il nous faut également faire le choix d’une base de données.

Nous allons tacher de mettre en place plusieurs bases de données, certains qu’on pourrait qualifier d’esclaves, elles seront sur les parties hors internet du projet. Une fois la connexion internet rétablies sur les machines possédants les bases locales, elles se synchroniseront avec la base dites Maitre qui elle se trouvera sur le réseau.

Notre choix de SGBD c’est porté sur MySQL car :

#### Rapide

Le serveur MySQL est très rapide. Des tests de performances sont disponibles sur le site de MySQL

#### Facile à utiliser

MySQL est beaucoup plus simple à utiliser que la plupart des serveurs de bases de données commerciaux.

#### API diverses

On peut effectuer diverses opérations sur une base MySQL en utilisant des interfaces écrites en C, Perl, C++, Java, Python, PHP.

#### Connexion et Sécurité

MySQL dispose d'un système de sécurité permettant de gérer les personnes et les machines pouvant accéder aux différentes bases.

#### Portabilité

MySQL tourne sur divers systèmes tels que Unix, Windows, Linux ou OS/2.



# Méthode de développement

Dans le cadre du projet, la méthode de développement choisie par l’équipe sera la méthode AGIL.

Nous allons procéder à des pokers pour définir les priorités des modules du projet, ainsi que des périodes de rush pour le développement. Des réunions régulières sur l’avancement du projet seront également mises en place pour avoir une bonne cohésion et lisibilité entre les membres l’équipe et leur travail associé, cela permettra aussi de prévenir d'éventuels blocages ou manques de connaissances.

La méthode AGIL nous permettra d’avancer le plus rapidement possible dans le projet tout en ayant une vision sur le travail de chacun et une connaissance de l’avancée dans chaque module du projet développé au même moment. Elle nous permettra également de pouvoir faire des erreurs et de modifier notre axe de développement.

